What is claimed is:

1. A compiler comprising:

a conversion program,

wherein the conversion program can convert first program descriptions described by diverting a predetermined program language into circuit descriptions,

the first program descriptions contain register assignment statements with particular operators and clock boundary descriptions, and which permit circuit operations to be specified at a cycle precision, and

the circuit descriptions specify hardware realizing the circuit operations specified by the first program descriptions in a predetermined hardware description language.

2. A compiler comprising:

a conversion program,

wherein the conversion program can convert first program descriptions described by diverting a predetermined program language can be converted into second program descriptions using a predetermined program language,

the first program descriptions contain register assignment statements with particular operators and clock boundary descriptions , and which permit circuit operations to be specified at a cycle precision, and

the second program descriptions contain transformed assignment statements into which the register assignment

statements are transformed in order to make states of preceding cycles referable, and register assignment description insertion statements which associate variables of the transformed assignment statements with changes of registers attendant upon cycle changes, in correspondence with the clock boundary descriptions.

3. A compiler comprising:

a conversion program,

wherein the conversion program can convert first program descriptions described by diverting a predetermined program language into second program descriptions using a predetermined program language,

the first program descriptions contain register assignment statements with particular operators and clock boundary descriptions, and which permit circuit operations to be specified at a cycle precision,

the second program descriptions contain transformed assignment statements into which the register assignment statements are transformed in order to make states of preceding cycles referable, and register assignment description insertion statements which associate variables of the transformed assignment statements with changes of registers attendant upon cycle changes, in correspondence with the clock boundary descriptions, and

the circuit descriptions specify hardware which is

defined by the second program descriptions, in a predetermined hardware description language.

4. The compiler of claim 1, wherein the predetermined program language is a C language.

5. The compiler of claim 1, wherein the hardware description language is a description language of RTL level.

6. A logic circuit design method comprising:

a first step; and

a second step,

wherein the first step inputs first program descriptions which contain register assignment statements and clock boundary descriptions bearing peculiar operators, which are described by diverting a predetermined program language in order to define circuit operations on the basis of timing specifications, and which permit the circuit operations to be specified at a cycle precision, and

the second step generats circuit information which satisfies the timing specifications, on the basis of the first program descriptions.

7. The logic circuit design method of claim 6, wherein the second step comprises the step of converting the first program descriptions, and generating as the circuit information, second program descriptions containing descriptions into which the register assignment statements are transformed using input variables and output variables, and

which assign the input variables to the output variables in correspondence with the clock boundary descriptions.

8. The logic circuit design method of claim 7, wherein the second step comprises the step of converting the second program descriptions, and generating as the further circuit information, circuit descriptions which serve to specify hardware satisfying the timing specifications, in a predetermined hardware description language.

9. The logic circuit design method of claim 8, wherein the program language is a C language.

10. The logic circuit design method of claim 9, further comprising the third step of performing a simulation of a circuit to-be-designed by employing the second program descriptions.

11. The logic circuit design method of claim 6, wherein the second step comprises the step of converting the first program descriptions, and generating as the circuit information, second program descriptions containing descriptions into which the register assignment statements are transformed using input variables and output variables.

12. The logic circuit design method of claim 11, wherein the second step comprises the step of converting the second program descriptions, and generating as the circuit information, third program descriptions containing descriptions which assign the input variables to the output

variables in correspondence with the clock boundary descriptions, and being described in a predetermined program language so as to be executable by a computer.

13. The logic circuit design method of claim 12, further comprising the third step of performing a simulation of a circuit to-be-designed by employing the third program descriptions.

14. The logic circuit design method comprising:

an input step; and

conversion step,

Wherein the input step inputs first program descriptions which contain register assignment statements and clock boundary descriptions bearing peculiar operators, which are described by diverting a predetermined program language in order to define circuit operations on the basis of timing specifications, and which permit the circuit operations to be specified at a cycle precision, and

the conversion step generates second program descriptions containing descriptions into which the register assignment statements are transformed using input variables and output variables and which assign the input variables to the output variable in correspondence with the clock boundary descriptions, and being described in the predetermined program language.

15. The logic circuit design method of claim 14, wherein,

in course of generating a CFG on the basis of the first program descriptions, the conversion step sets clock boundary nodes in the CFG in correspondence with the clock boundary descriptions, whereupon it inserts the register assignment descriptions behind the clock boundary nodes.

16. The logic circuit design method of claim 15, further comprising an optimization step of optimizing codes of the second program descriptions, while a variable table of respective state transitions is being created by utilizing the CFG.

17. The logic circuit design method of claim 16, further comprising a "retain" step of extracting parts in which variables do not change between states within the variable table, as parts which need to be retained, and adding descriptions for assigning the input variables to the output variables, to the extracted parts.

18. The logic circuit design method of claim 17, further comprising an extraction step of extracting the codes which constitute state machines, on the basis of the variables and arguments of the respective state transitions within the variable table having undergone said "retain" step.

19. The logic circuit design method of claim 18, further comprising the step of describing hardware of a circuit which satisfies the circuit specifications, in a predetermined hardware description language, while reference is being had

to the state machine constituting codes extracted by said extraction step, and to the second program descriptions.

20. The logic circuit design method of claim 14, wherein, whether or not a loop to be executed in the 0th cycle is existent is decided for the first program descriptions, and when the loop has been decided to be nonexistent, the step of describing hardware of a circuit which satisfies the circuit specifications, in a predetermined hardware description language, is performed.